ESTF2021
Virtual Series

New Observing Strategies Testbed (NOS-T) Design and Development

June 10, 2021

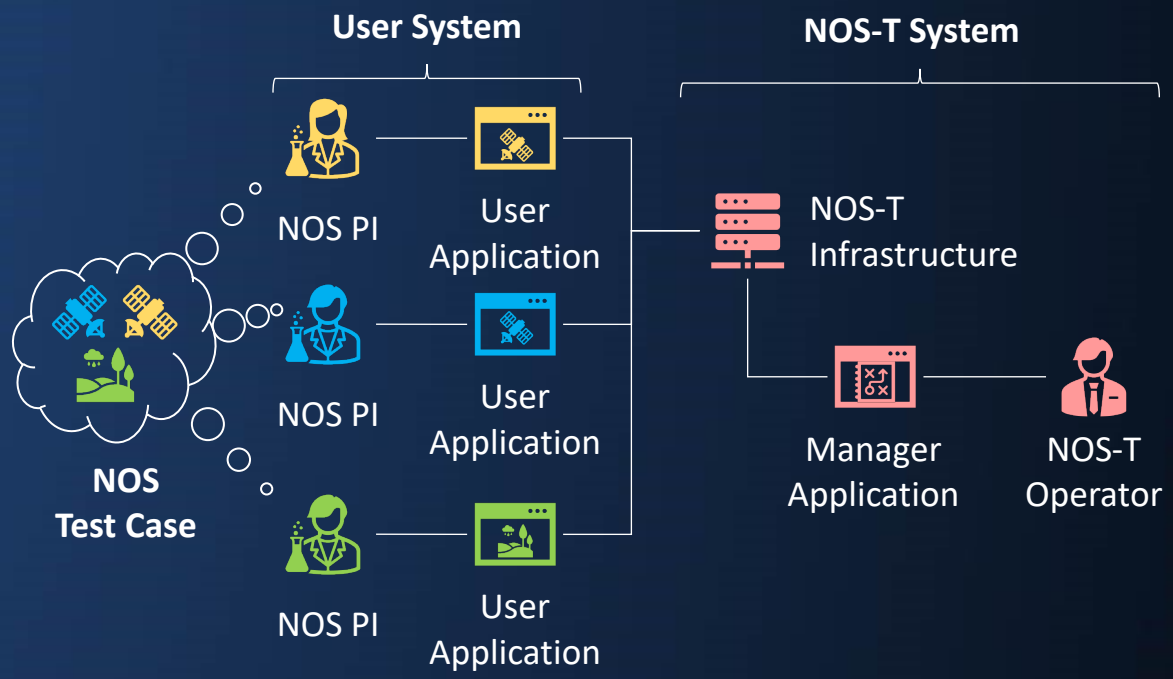
Paul Grogan, Ph.D.
Jerry Sellers, Ph.D.
Matthew Brand
Hayden Daly

Systems Engineering Research Center

Stevens Institute of Technology

NOS Testbed (NOS-T)

- **NOS-T** is a computational platform to prototype and mature NOS concepts
- **NOS-T** Objectives:
  - Validate NOS technologies, independently and as a system
  - Demonstrate novel distributed operational concepts
  - Enable meaningful comparisons of competing technologies
  - Socialize new technologies and concepts with the science community by retiring the risk of integration
- **NOS-T Framework** Objectives:
  - Enable disparate organizations to propose and participate in developing NOS technology

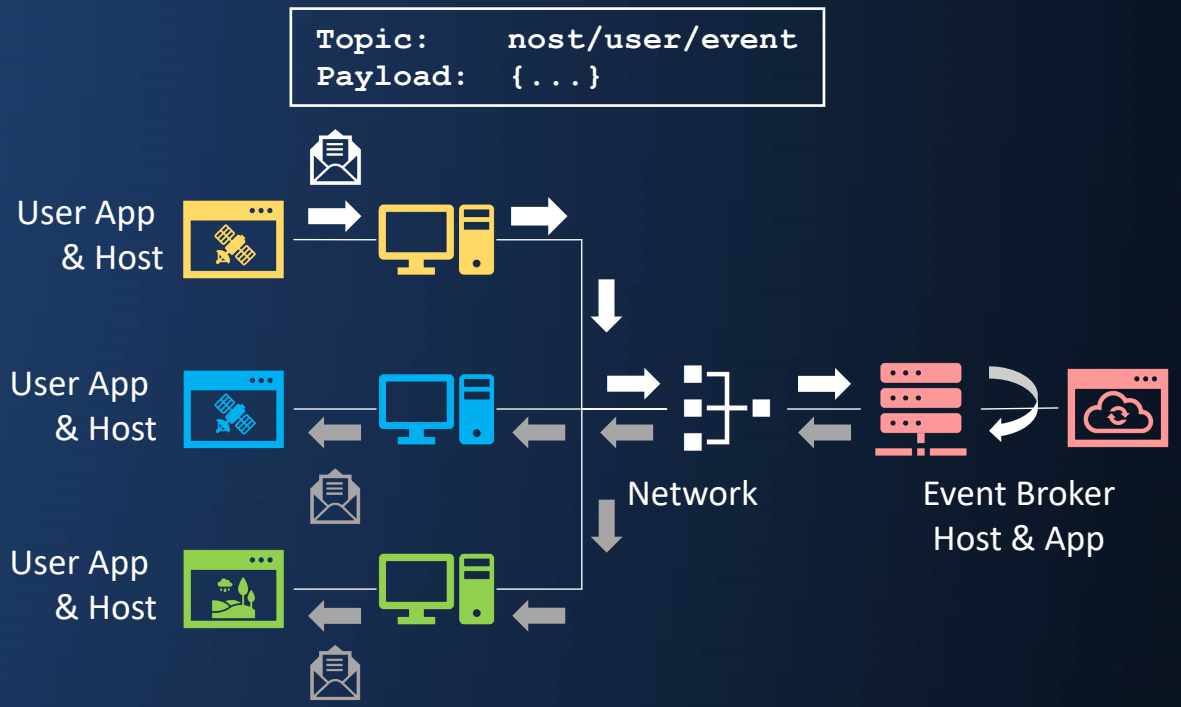


NOS-T Framework Principles

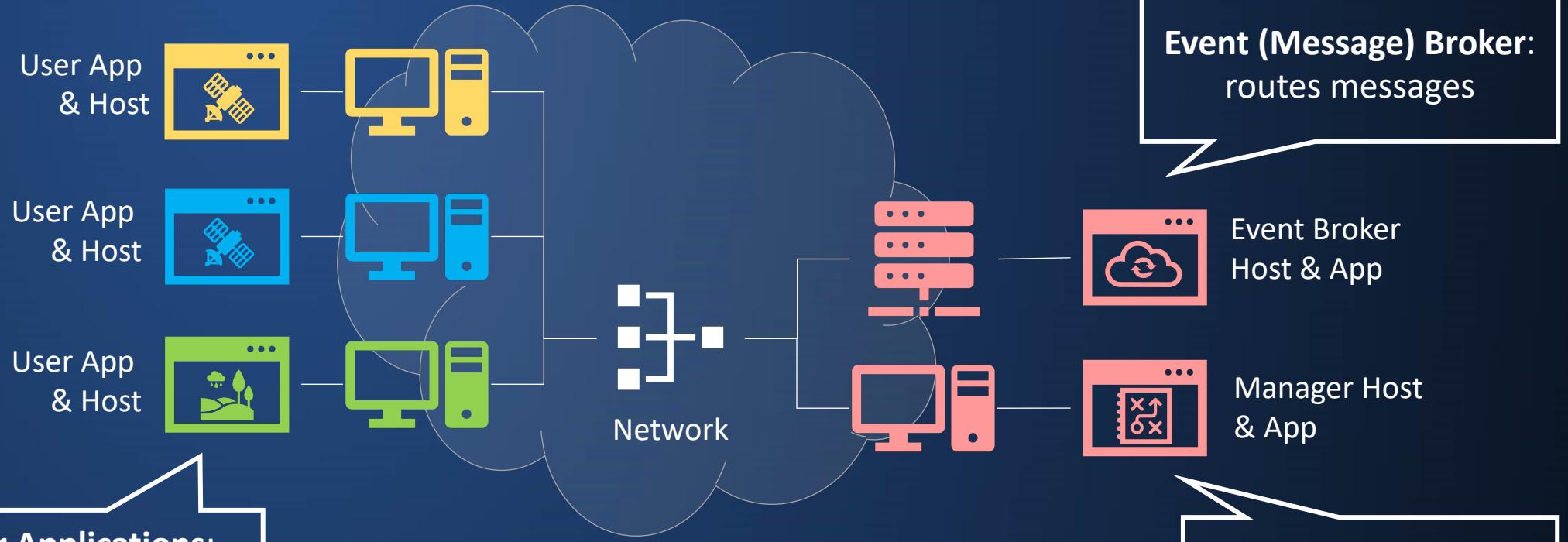
- **Geographic distribution**: user applications interconnect using standard network interfaces
- **Multi-party participation**: user applications exchange limited information via standard network protocols
- **Security**: encrypt transport data, allow fine-grain access control rules, monitor hosted infrastructure on authorized information systems
- **Modularity**: loose coupling allows user applications to be added and updated without modifying the testbed
- **Extensibility**: vary the number or capabilities of user applications to explore a wide range of test cases
- **Usability**: allow members of the Earth science community to develop test cases and user applications without a substantial learning curve

NOS-T System Architecture

- Event-driven architecture (EDA):
  - Applications communicate state changes through *events*
  - Event (message) broker routes events as notification messages
- NOS-T Broker: Solace PubSub+
  - Standard edition supports up to 1000 concurrent connections and 10,000 messages/second
  - Multiple messaging protocols
  - Hosted on Science Managed Cloud Environment (SMCE), a FISMA Low information system

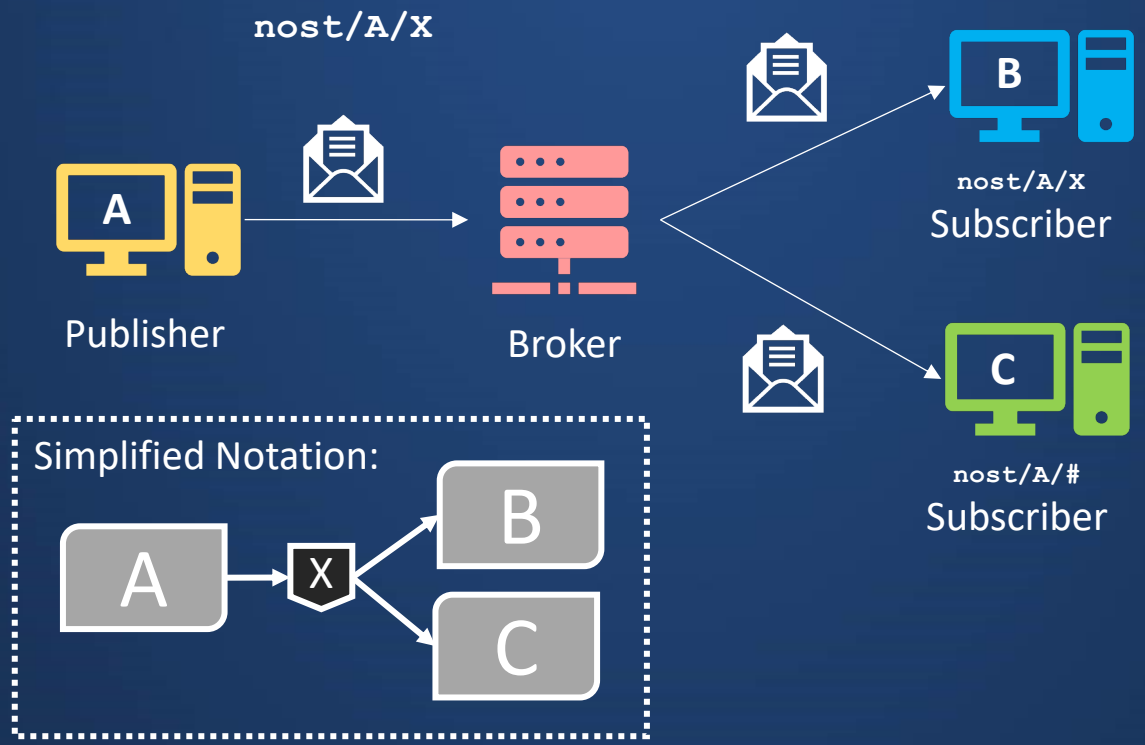


NOS-T System Components



NOS-T System Interfaces

- Messaging protocol with publish-subscribe pattern



- Recommend hierarchical topic addressing scheme

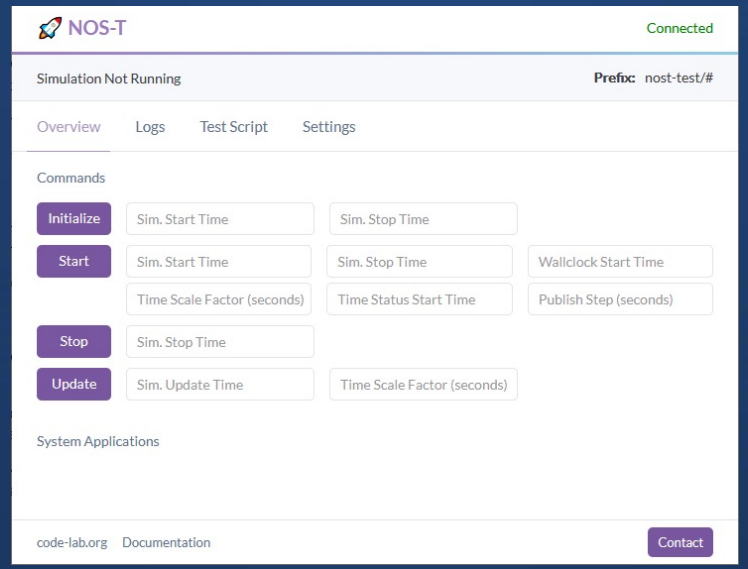
`test-case/app/event` e.g., `nost/manager/start`

- Recommend encoding message payload as JSON
  - Leverage standard object schemas, e.g., SensorThings

```
{
  "taskingParameters": {
    "startTime": "2021-04-15T12:00",
    "simStartTime": "2019-03-15T00:00",
    "simStopTime": "2019-03-19T00:00",
    "timeScalingFactor": 60
  }
}
```

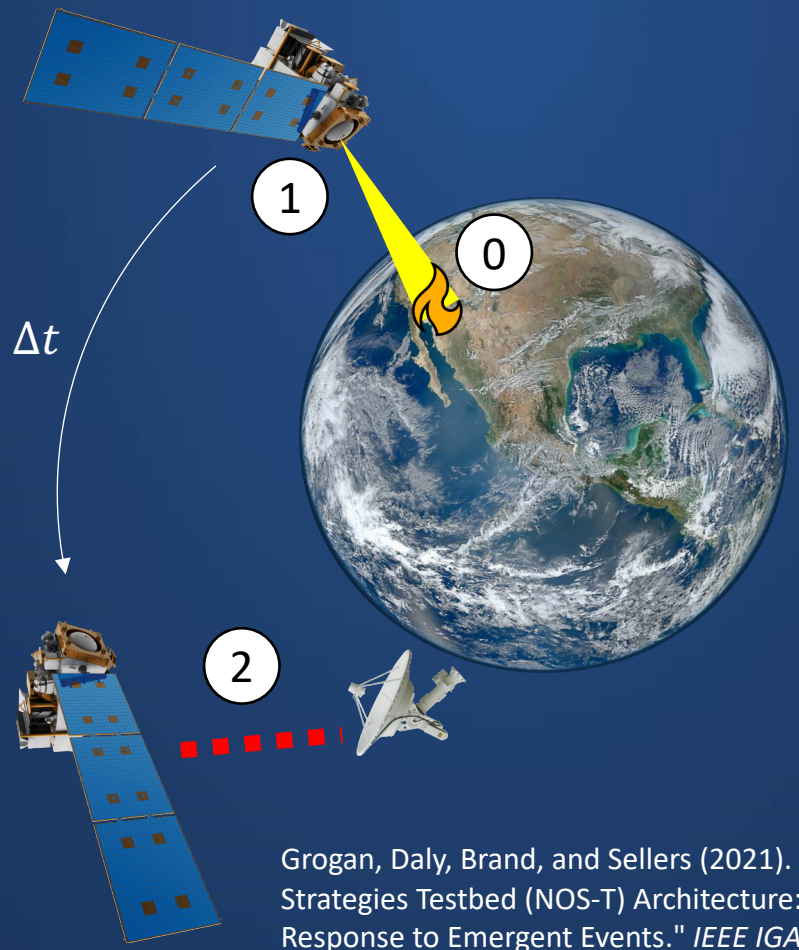
NOS-T Manager Events

- The manager application orchestrates test run executions
  - Backend: precision timing loop
  - Frontend: operator interface
- Publishes events for user apps



- **Control events**: task a user application in the test run
  - **Initialize**: set scenario start/end
  - **Start**: schedule start of test run
  - **Update**: modify test run options
  - **Stop**: schedule end of test run
- **Status events**: communicate test run state changes
  - **Time**: periodically send the current scenario time
  - **Mode**: initializing, initialized, executing, terminating, terminated

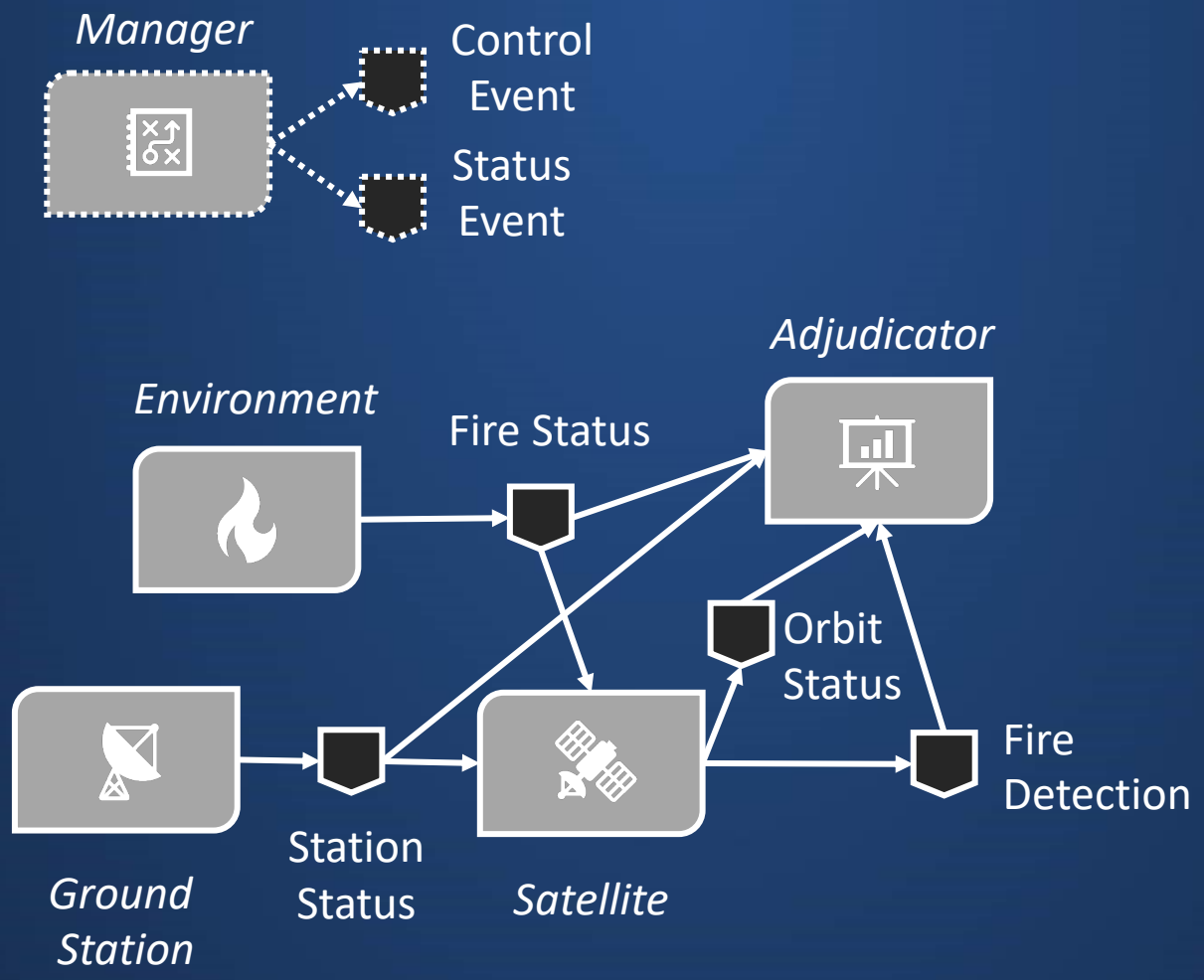
Example Test Case: FireSat



- Remote fire hazard detection in continental U.S.
  - Fire initiation and growth
  - Remote observation by single sun-synchronous satellite
  - Data downlink to ground station
  - Evaluate of key performance measures (observation latency)
- Extensible to design-of-experiment studies to assess observing system variables

Grogan, Daly, Brand, and Sellers (2021). "New Observing Strategies Testbed (NOS-T) Architecture: Evaluating Dynamic Response to Emergent Events." *IEEE IGARSS 2021*. Accepted.

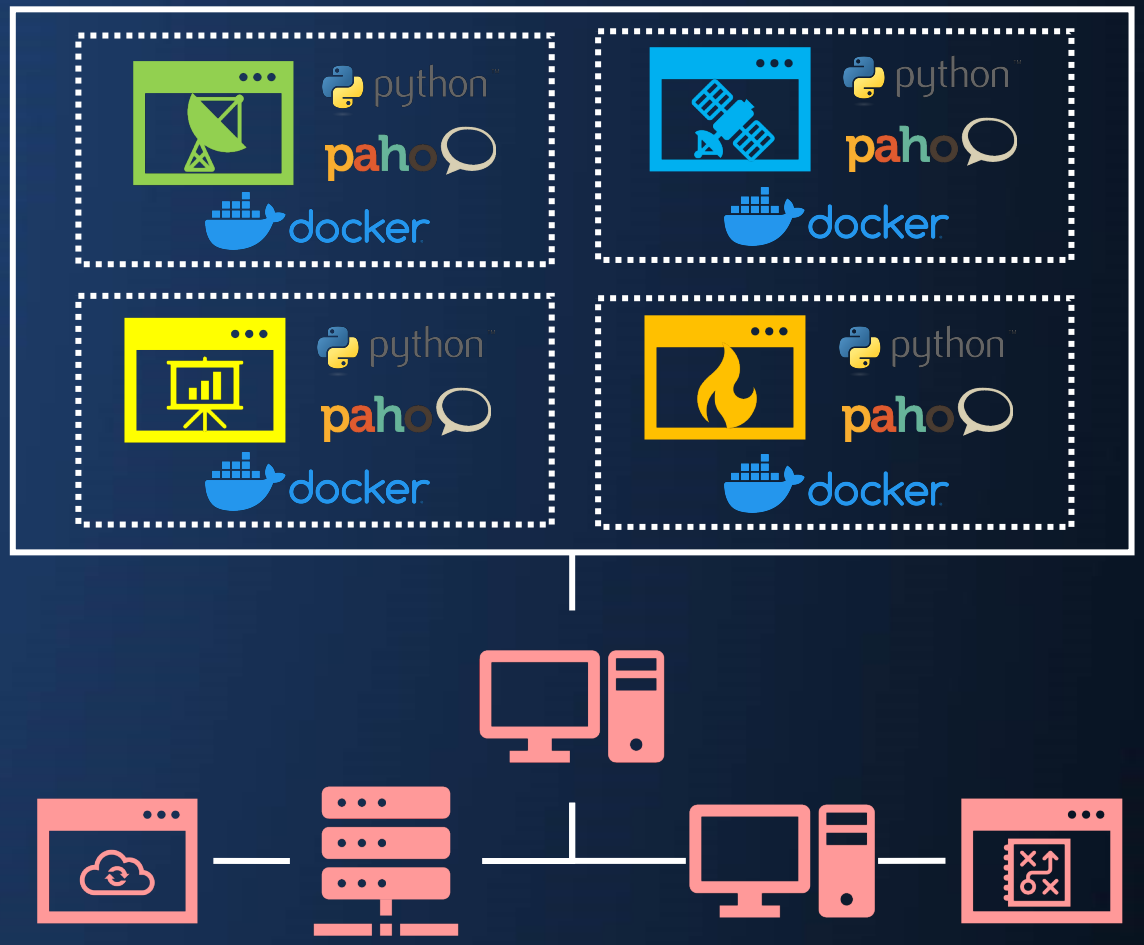
User Applications & Events



- **Environment**: models fire ignition and growth
- **Ground Station**: models ground station operation
- **Satellite**: models orbit propagation and detection
- **Adjudicator**: visualizes and logs data to compute key performance measures

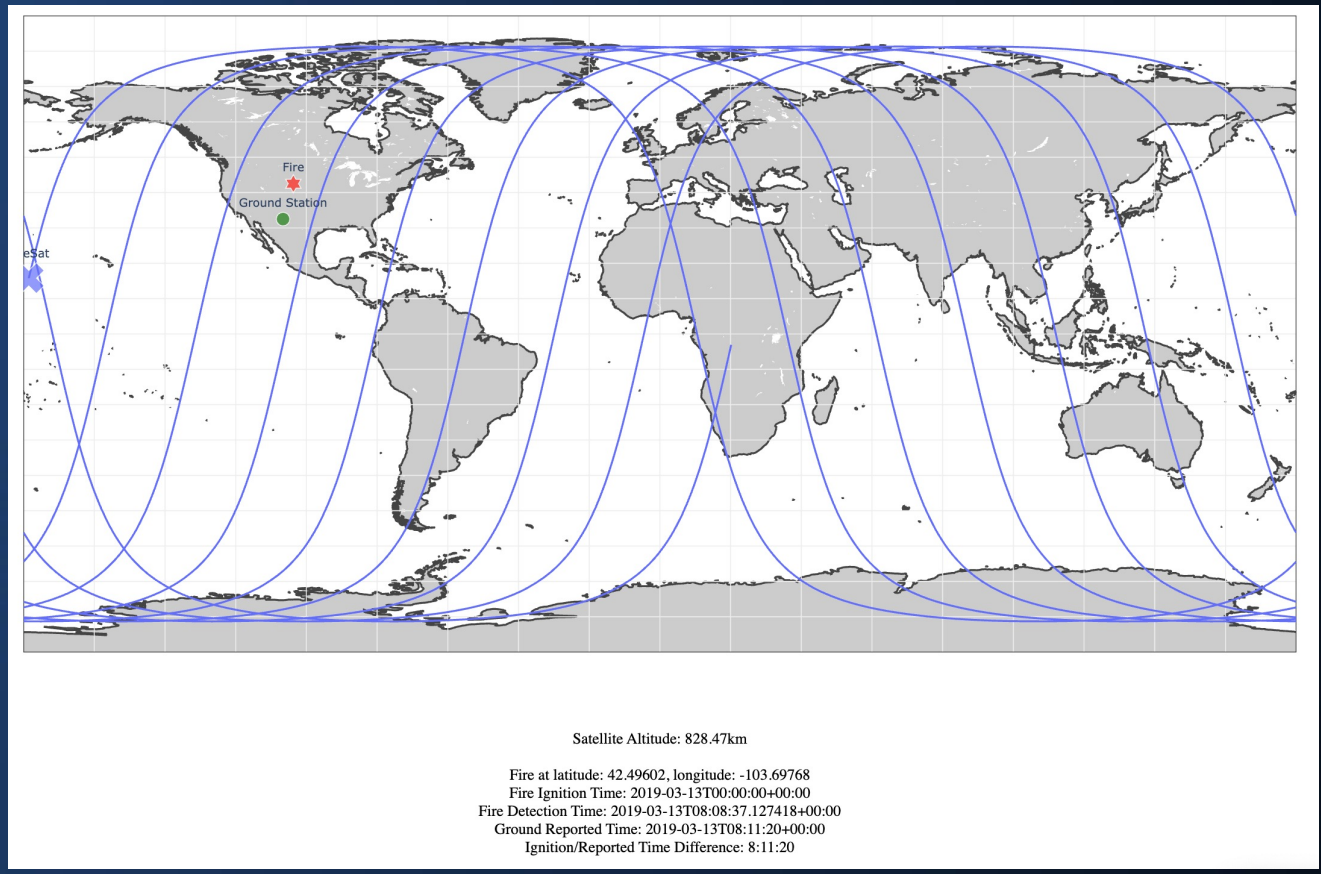
Test Case Execution

- User applications implemented using Python and Eclipse Paho MQTT
- Applications containerized using Docker virtualization
- Applications launched using docker-compose
- Connect to an event broker hosted on SMCE and subscribe to manager events

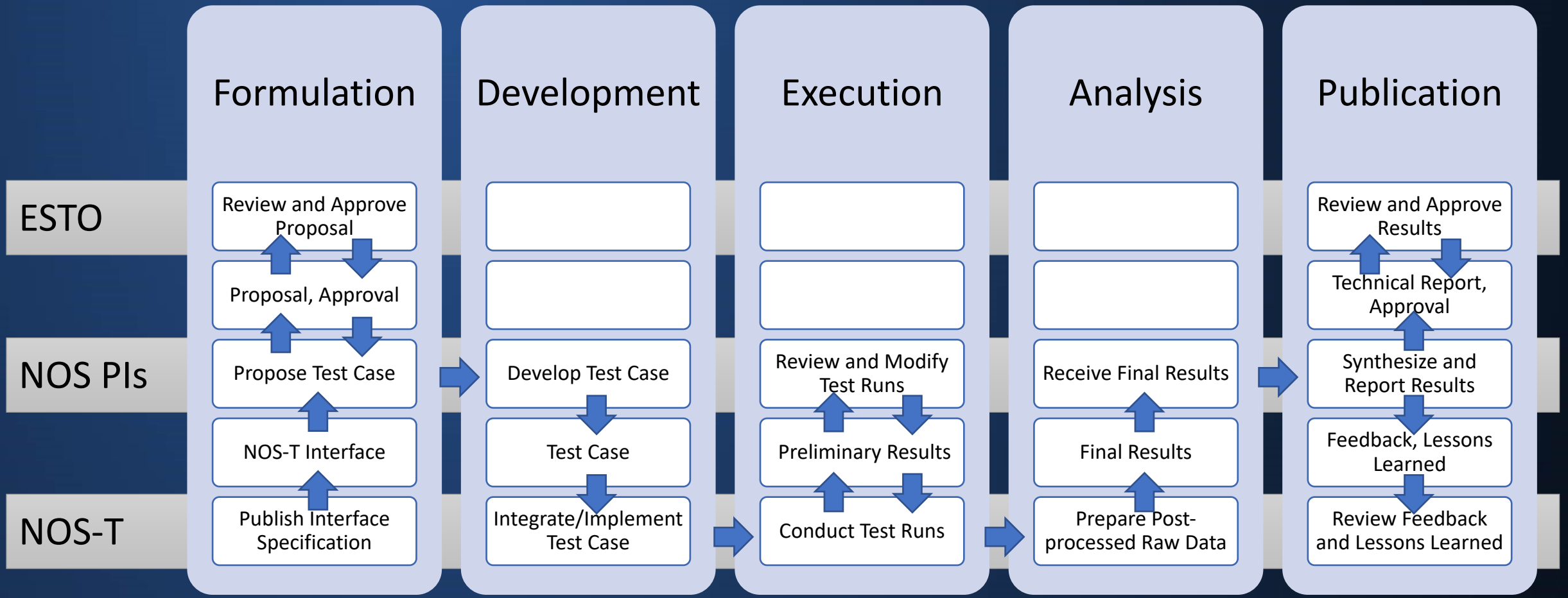


Test Case Visualization

- Adjudicator application runs a web-based dashboard to visualize scenario execution
  - Fire location(s)
  - Ground station location(s)
  - Satellite ground track
- Computes and reports detection latency as a key performance metric



NOS-T Test Case ConOps



NOS-T Framework Status

- "NOS-T Design and Development" started in August 2020
- Version 1.0 (alpha) specification completed in May 2021
- Version 1.0 release scheduled for February 2022
- Version 2.0 design and development through August 2023

Acknowledgements

This material is based upon work supported, in whole or in part, by the U.S. Department of Defense through the Office of the Assistant Secretary of Defense for Research and Engineering (ASD(R&E)) under Contract W15QKN-18-D-0040, Task Order W15QKN-20-F-0051, sponsored by the National Aeronautics and Space Administration's Earth Science Technology Office.

Thanks to the NASA ESTO representatives and NOS pilot project teams for close collaboration and regular feedback.